

# Guía de inicio rápido en FreeBSD para usuarios de Linux

Resumen

El objetivo de este documento es familiarizar rápidamente a los usuarios intermedios y avanzados de Linux® con los conceptos básicos de FreeBSD.

---

## Tabla de contenidos

1. Introducción .....	1
2. Shell por defecto .....	1
3. Paquetes y Ports: Instalar software en FreeBSD .....	2
4. Inicio del sistema .....	3
5. Configuración de la red .....	4
6. Firewall .....	5
7. Actualizando FreeBSD .....	6
8. procs: Desaparecido pero no olvidado .....	6
9. Comandos comunes .....	7
10. Conclusión .....	7

## 1. Introducción

Este documento destaca algunas de las diferencias técnicas entre FreeBSD y Linux® para que los usuarios intermedios y avanzados de Linux® puedan familiarizarse rápidamente con los conceptos básicos de FreeBSD.

This document assumes that FreeBSD is already installed. Refer to the [Installing FreeBSD](#) chapter of the FreeBSD Handbook for help with the installation process.

## 2. Shell por defecto

Los usuarios de Linux® a menudo se sorprenden al descubrir que Bash no es la shell por defecto en FreeBSD. De hecho, Bash no está incluido en la instalación predeterminada. En su lugar, FreeBSD utiliza `tcsh(1)` como shell predeterminada para el usuario root y `sh(1)` como shell compatible con Bourne shell por defecto. `sh(1)` es muy similar a Bash pero con un conjunto de características mucho más pequeño. Generalmente, los scripts escritos para `sh(1)` se ejecutarán en Bash, pero al contrario no siempre es así.

Sin embargo, Bash y otras shells están disponibles para la instalación utilizando los [paquetes de FreeBSD y la Colección de Ports](#).

---

Después de instalar otra shell, use `chsh(1)` para cambiar la shell predeterminada de un usuario. Se recomienda mantener la shell del usuario `root`, ya que las shells que no están incluidas en el sistema base se instalan en `/usr/local/bin`. Si hay un problema, el sistema de archivos donde se encuentra `/usr/local/bin` podría no estar montado. En este caso, el usuario `root` podría no tener acceso a su shell por defecto, impidiendo que el usuario `root` inicie sesión y solucione el problema.

## 3. Paquetes y Ports: Instalar software en FreeBSD

FreeBSD proporciona dos métodos para instalar aplicaciones: paquetes binarios y ports compilados. Cada método tiene sus propias ventajas:

### *Paquetes binarios*

- Instalación más rápida comparado con la compilación de aplicaciones de gran tamaño.
- No es necesario saber cómo compilar software.
- No es necesario instalar un compilador.

### *Ports*

- Posibilidad de personalizar las opciones de instalación.
- Se pueden aplicar parches personalizados.

Si la instalación de una aplicación no requiere de ninguna personalización, con la instalación del paquete es suficiente. Compile el port siempre que una aplicación requiera la personalización de las opciones predeterminadas. Si fuera necesario, se puede compilar un paquete personalizado desde los ports usando `make `package``.

Una lista completa de todos los ports y paquetes disponibles se puede encontrar [aquí](#).

### 3.1. Paquetes

Packages are pre-compiled applications, the FreeBSD equivalents of `.deb` files on Debian/Ubuntu based systems and `.rpm` files on Red Hat/Fedora based systems. Packages are installed using `pkg`. For example, the following command installs Apache 2.4:

```
# pkg install apache24
```

Para obtener más información sobre los paquetes, consulte la sección 5.4 del Manual de FreeBSD: [Uso de pkgng para la administración de paquetes binarios](#).

### 3.2. Ports

La Colección de Ports de FreeBSD es un framework de Makefiles y parches específicamente personalizados para instalar aplicaciones con su código fuente en FreeBSD. Al instalar un port, el sistema buscará el código fuente, aplicará los parches necesarios, compilará el código e instalará la

aplicación y las dependencias necesarias.

La Colección de Ports, a veces llamada el árbol de ports, se puede instalar en `/usr/ports` utilizando [portsnap\(8\)](#). Se pueden encontrar instrucciones detalladas para instalar la Colección de Ports en la [sección 5.5](#) del Manual de FreeBSD.

Para compilar un port, acceda al directorio del port e inicie el proceso de build. El siguiente ejemplo instala Apache 2.4 de la colección de ports:

```
# cd /usr/ports/www/apache24
# make install clean
```

El beneficio de usar ports para instalar software es la capacidad de personalizar las opciones de instalación. Este ejemplo especifica que el módulo `mod_ldap` también debe ser instalado:

```
# cd /usr/ports/www/apache24
# make WITH_LDAP="YES" install clean
```

Consulte [Usando la Colección de Ports](#) para obtener más información.

## 4. Inicio del sistema

Muchas distribuciones de Linux® usan el sistema de inicio SysV, mientras que FreeBSD usa el tradicional BSD-style [init\(8\)](#). Con el BSD-style [init\(8\)](#), no hay run-levels y `/etc/inittab` no existe. En su lugar, el inicio está controlado por scripts [rc\(8\)](#). En el arranque del sistema, `/etc/rc` lee `/etc/rc.conf` y `/etc/defaults/rc.conf` para determinar qué servicios deben iniciarse. Los servicios especificados son iniciados ejecutando su correspondiente script de inicialización del servicio ubicado en `/etc/rc.d/` y `/usr/local/etc/rc.d/`. Estos scripts son similares a los scripts ubicados en `/etc/init.d/` en los sistemas Linux®.

Los scripts ubicados en `/etc/rc.d/` son para las aplicaciones que forman parte del sistema "base", como [cron\(8\)](#), [sshd\(8\)](#), y [syslog\(3\)](#). Los scripts ubicados en `/usr/local/etc/rc.d/` son para aplicaciones instaladas por el usuario como Apache y Squid.

Dado que FreeBSD se desarrolla como un sistema operativo completo, las aplicaciones instaladas por el usuario no se consideran parte del sistema "base". Las aplicaciones instaladas por el usuario generalmente se instalan utilizando [Paquetes o Ports](#). Para mantenerlas separadas del sistema base, las aplicaciones instaladas por el usuario se instalan en `/usr/local/`. Por lo tanto, los binarios instalados por el usuario están ubicados en `/usr/local/bin/`, los archivos de configuración están en `/usr/local/etc/`, y así sucesivamente.

Los servicios se habilitan añadiendo una entrada para el servicio en `/etc/rc.conf`. Los valores predeterminados del sistema se encuentran en `/etc/defaults/rc.conf` y las configuraciones por defecto se sobrescriben con `/etc/rc.conf`. Consulte [rc.conf\(5\)](#) para obtener más información sobre las entradas disponibles. Al instalar aplicaciones adicionales, revise el mensaje de instalación para determinar cómo habilitar los servicios asociados.

Las siguientes entradas en `/etc/rc.conf` activan [sshd\(8\)](#), activan Apache 2.4 y especifican que Apache debe iniciarse con SSL.

```
# activa SSHD
sshd_enable="YES"
# activa Apache con SSL
apache24_enable="YES"
apache24_flags="-DSSL"
```

Una vez que un servicio ha sido activado en `/etc/rc.conf`, puede iniciarse sin reiniciar el sistema:

```
# service sshd start
# service apache24 start
```

Si un servicio no ha sido activado, puede iniciarse desde la línea de comandos usando [onstart](#):

```
# service sshd onstart
```

## 5. Configuración de la red

Instead of a generic *ethX* identifier that Linux® uses to identify a network interface, FreeBSD uses the driver name followed by a number. The following output from [ifconfig\(8\)](#) shows two Intel® Pro 1000 network interfaces (em0 and em1):

```
% ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM, TXCSUM, VLAN_MTU>
    inet 10.10.10.100 netmask 0xffffffff broadcast 10.10.10.255
    ether 00:50:56:a7:70:b2
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM, TXCSUM, VLAN_MTU>
    inet 192.168.10.222 netmask 0xffffffff broadcast 192.168.10.255
    ether 00:50:56:a7:03:2b
    media: Ethernet autoselect (1000baseTX <full-duplex>)
    status: active
```

Una dirección IP puede ser asignada a una interfaz usando [ifconfig\(8\)](#). Para que permanezca entre reinicios, la configuración IP debe ser incluida en `/etc/rc.conf`. Las siguientes entradas en `/etc/rc.conf` especifican el hostname, la dirección IP, y el gateway por defecto:

```
hostname="server1.example.com"
ifconfig_em0="inet 10.10.10.100 netmask 255.255.255.0"
defaultrouter="10.10.10.1"
```

En su lugar, utilice las siguientes entradas para configurar una interfaz de red con DHCP:

```
hostname="server1.example.com"
ifconfig_em0="DHCP"
```

## 6. Firewall

FreeBSD no usa las IPTABLES de Linux® para su firewall. En su lugar, FreeBSD ofrece tres firewalls a nivel del kernel:

- [PF](#)
- [IPFILTER](#)
- [IPFW](#)

PF está desarrollado por el proyecto OpenBSD y portado a FreeBSD. PF fue creado como un reemplazo para IPFILTER y su sintaxis es similar. PF se puede combinar con [altq\(4\)](#) para proporcionar QoS.

Este ejemplo de PF permite la entrada de tráfico SSH:

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port 22
```

IPFILTER es el firewall desarrollado por Darren Reed. No es específico de FreeBSD y se ha portado a varios sistemas operativos, incluidos NetBSD, OpenBSD, SunOS, HP/UX y Solaris.

La sintaxis de IPFILTER para permitir la entrada de tráfico SSH es:

```
pass in on $ext_if proto tcp from any to any port = 22
```

IPFW es el cortafuegos desarrollado y mantenido por FreeBSD. Se puede combinar con [dummynet\(4\)](#) para proporcionar traffic shaping y simular diferentes tipos de conexiones.

La sintaxis de IPFW para permitir la entrada de tráfico SSH sería:

```
ipfw add allow tcp from any to me 22 in via $ext_if
```

## 7. Actualizando FreeBSD

Hay dos métodos para actualizar un sistema FreeBSD: desde el código fuente o desde la actualización de los binarios.

Actualizar desde código fuente es el método más complejo pero el que ofrece mayor flexibilidad. El proceso implica la sincronización de una copia local del código fuente de FreeBSD con los servidores Subversion de FreeBSD. Una vez actualizado el código fuente, puede compilar nuevas versiones del kernel y utilidades.

Las actualizaciones de los binarios son similares a usar `yum` o `apt-get` para actualizar un sistema Linux®. En FreeBSD, `freebsd-update(8)` puede usarse para obtener las nuevas actualizaciones de los binarios e instalarlas. Estas actualizaciones pueden ser programadas usando `cron(8)`.



Cuando use `cron(8)` para programar actualizaciones, use `freebsd-update cron` en `crontab(1)` para reducir la posibilidad de que una gran cantidad de máquinas se actualicen al mismo tiempo:

```
0 3 * * * root /usr/sbin/freebsd-update cron
```

Para obtener más información de las actualizaciones de código y binarias, consulte el [capítulo sobre la actualización](#) en el Manual de FreeBSD.

## 8. procfs: Desaparecido pero no olvidado

En algunas distribuciones de Linux®, puede consultar `/proc/sys/net/ipv4/ip_forward` para determinar si IP forwarding está habilitado. En FreeBSD, `sysctl(8)` se usa para ver esta y otras configuraciones del sistema.

Por ejemplo, utilice el siguiente comando para comprobar si IP forwarding está habilitado en FreeBSD:

```
% sysctl net.inet.ip.forwarding
net.inet.ip.forwarding: 0
```

Use `-a` para listar todos los ajustes del sistema:

```
% sysctl -a | more
```

Si una aplicación necesita procfs, añada la siguiente línea a `/etc/fstab`:

```
proc                /proc                procfs  rw,noauto            0                0
```

Incluir `noauto` evitará que `/proc` se monte automáticamente en el arranque.

Para montar el sistema de archivos sin reiniciar:

```
# mount /proc
```

## 9. Comandos comunes

Algunos equivalentes de los comandos comunes son los siguientes:

Linux command (Red Hat/Debian)	Equivalente en FreeBSD	Objetivo
<code>yum install package</code> / <code>apt-get install package</code>	<code>pkg install package</code>	Instalar el paquete desde el repositorio remoto
<code>rpm -ivh package</code> / <code>dpkg -i package</code>	<code>pkg add package</code>	Instalar un paquete local
<code>rpm -qa</code> / <code>dpkg -l</code>	<code>pkg info</code>	Listar los paquetes instalados
<code>lspci</code>	<code>pciconf</code>	Listar los dispositivos PCI
<code>lsmod</code>	<code>kldstat</code>	Listar los módulos cargados en el kernel
<code>modprobe</code>	<code>kldload</code> / <code>kldunload</code>	Cargar/Descargar módulos del kernel
<code>strace</code>	<code>truss</code>	Rastrear llamadas al sistema

## 10. Conclusión

This document has provided an overview of FreeBSD. Refer to the [FreeBSD Handbook](#) for more in-depth coverage of these topics as well as the many topics not covered by this document.